REMARKS

Claims 27-35 and 38-51 stand rejected under § 103 on the basis of Faraboschi et al. '576. The claims have been amended to overcome this rejection, which applicant traverses for the following reasons.

The independent claims have been amended to take account of the points made by the examiner in paragraphs 45 to 49 of the office action. Each of the independent claims now recites that the program memory in which the compressed program is stored is *outside the processor*. The imaginary addresses assigned to the original instructions are now said to *determine at execution time respective locations in the instruction cache into which instructions read from the program memory are to be loaded after decompression.* Accordingly, the purpose of the imaginary address information is now defined more concretely in the independent claims.

The new independent claims also have been clarified to recite how the compressed-form instructions and the imaginary address information are read by the processor. It is now recited that at execution time the processor reads both the compressed-form instructions and the imaginary address information from *program-memory addresses pointed to directly by the program counter*. It is also indicated that the processor is able to load the compressed instructions into instruction-cache locations *determined by* the allocated imaginary addresses.

In Faraboschi, the main memory 110, comprising the code pointer segment 130 and code heap segment 140, can be considered to be a program memory outside the

processor. Similarly, the instruction cache 100 is inside the processor. The code heap segment 140 stores exclusively compressed instructions. The code pointer segment 130 stores a series of code pointer words 132, 134, 136. Each code pointer word comprises a nine-bit mask, which is similar to the decompression key of the present invention, and a 9-bit code pointer (ptr) 152. The code pointer 152 represents an offset between the address in the program memory (code pointer segment 130) at which the code pointer word containing the code pointer is stored and the actual address in the program memory (code heap segment 140) in which the first compressed-form instruction is stored. For example, in Fig. 2 of Faraboschi, the code pointer word 132 stored at program-memory address 12345100 comprises a code pointer 152 of +12ec80. When this offset is added to the program-memory address 12345100 of the code pointer word 132 the program-memory address 14000300 is attained, and this is the actual address at which the required compressed-form instruction W00 is stored.

For the reasons set out below, Faraboschi cannot fairly be said to meet the requirements of the independent claims as now amended.

Firstly, Faraboschi does not read imaginary address information stored *outside the processor*. It is true that, in Faraboschi, the code pointer segment 130 is outside the processor and that the program counter points to successive addresses in the code pointer segment 130 where code pointers are stored. However, Faraboschi's code pointer does not constitute imaginary address information. Faraboschi's code pointers 152 tell the processor, which generates *imaginary* addresses with its program counter, what are the *real* addresses at

which the compressed-form instructions corresponding to the internally-generated imaginary addresses will be found. The imaginary address information originates inside the processor in Faraboschi; it is the real addresses that are found outside using the code pointers.

Secondly, the independent claims as now amended require that at execution time the imaginary addresses that were assigned to the compressed-form instructions at the time of compression are allocated by the processor to the decompressed instructions *based on the imaginary address information* read from the program memory, and that decompressed instructions are loaded into instruction-cache locations *determined by the allocated imaginary addresses.* In other words, it is the externally-accessed imaginary address information that, at execution time, brings about the allocation of the previously-assigned imaginary addresses to the decompressed instructions and determines the instruction-cache locations of the decompressed instructions. In Faraboschi, the approach is different: the instruction-cache locations are determined directly by the program counter, which counts through imaginary addresses, and the code pointers have no influence over the instruction-cache locations of the decompressed instructions. Storing the imaginary address information externally is advantageous from the point of flexibility. For example, the imaginary addresses can be chosen (at the time of compressing the program) in such a way as to avoid mapping to the same cache block at execution tie instructions that are likely to coexist in the instruction cache (see claim 2).

Thirdly, in Faraboschi, the program counter of the processor does not point directly to the program-memory addresses at which the compressed-form instructions are

stored, and accordingly the compressed-form instructions cannot be *read from program-memory addresses pointed to directly by the program counter*, as required by the independent claims as now amended. Instead, the program counter in Faraboschi points to the program-memory addresses within the code pointer segment 130, i.e., it points directly to the code pointer words 132, 134, 136 etc. To access the compressed-form instructions, the code pointer words must be read first and the relevant offsets in the code pointers added to the program counter to arrive at the required program-memory address at which the compressed-form instruction is stored. This inevitably introduces an undesirable delay in accessing the compressed-form instruction.

The present invention is concerned with "on-the-fly" decompression of the compressed-form instructions. It is important to access the instructions as quickly as possible, since the accessing and subsequent decompression is a critical activity for the processor. In the present invention, the presence of the imaginary address information, together with the compressed-form instructions, and the use of the program counter to point directly to the compressed-form instructions and imaginary address information, enables the compressed instructions to be read from the program memory with minimum delay and enables the appropriate imaginary addresses (determining the instruction-cache locations of the decompressed instructions) to be determined as quickly as possible. These features are not taught or suggested by Faraboschi. Withdrawal of this rejection is respectfully requested.

For the foregoing reasons, applicants believe that this case is in condition for allowance, which is respectfully requested. The examiner should call applicants' attorney if an interview would expedite prosecution.

Respectfully submitted,

GREER, BURNS & CRAIN, LTD.

By 

Patrick G. Burns
Registration No. 29,367

March 14, 2006

300 South Wacker Drive
Suite 2500
Chicago, Illinois 60606
Telephone: 312.360.0080
Facsimile: 312.360.9315

Customer No. 24978